

Windows operating system malware detection using machine learning

Rawabi Hilabi, Ahmed Abu-Khadrah

College of Computing and Informatics, Saudi Electronic University, Riyadh, Saudi Arabia

Article Info

Article history:

Received Dec 12, 2023
Revised Feb 19, 2024
Accepted Mar 20, 2024

Keywords:

Extreme gradient boosting
Malicious
Malwares
Portable executable
Random forest

ABSTRACT

Over the years, cybercriminals have become more sophisticated in manipulating network users. Malware is a popular tool they use to exploit victims, targeting valuable assets such as identities and credit cards in the realm of digital technology. Cybersecurity professionals are consistently innovating to detect malicious activities. Machine learning (ML) algorithms are now a leading method for rapidly identifying unseen malware, offering efficiency and intelligence beyond traditional approaches. In fact, attackers like to see the victims suffer from damage caused by malware. Malware can destroy devices and networks. Additionally, hackers can blackmail individuals and organizations to obtain money through ransomware. Therefore, the aim of this research is developing a new model that has the capability of detecting malwares that are targeting Windows operating systems (OS) through enhancing an existing model by deploying several ML algorithms which are extreme gradient boosting (XGB) and random forest (RF). In addition, the swarm optimization and ML applied to portable executable (SOMLAP) dataset applied in the portable executable (PE) is used for training data and testing these learning algorithms. The result achieved by XGB and RF hybrid technique accuracy was 0.966, precision 0.990 and recall was 0.918.

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Ahmed Abu-Khadrah
College of Computing and Informatics, Saudi Electronic University
Riyadh, Saudi Arabia
Email: abosuliman2@yahoo.com

1. INTRODUCTION

Over decades technology is in growth status, criminals utilize malicious software to commit crimes digitally rather than committing crimes in the real world because attackers have the ability to impersonate themselves and hide their geographical locations and that makes it difficult to be caught. In fact, there are different types of malwares where each one of them is being deployed to achieve a certain objective. Malware is defined as any type of software or program that has the ability of delivering or executing payload against the targeted device like laptop, smartphone, and smartwatch [1]. As an example, spyware is a one of popular malwares that can allow the attacker to obtain information such as credit card information. Ransomware, the attacker aims to encrypt a file, so the users will not be able to read the data in the encrypted file. The objective of designing malware is to cause damage and chaos to the victim through stealing credentials, destroying the victim's device, and encrypting victim's data. In fact, hackers continuously increase the complexity of the malware and always aim to identify systems' flaws to create new malware. First of all, Windows users have to understand the popular types of malwares that can impact the devices [2]–[5].

Virus is a popular type of malware that has the ability to replicate itself and propagate itself on other computers. The ways that can be transferred are through files, media files, or network files. Based on the complexity of the code, it can change the replicated copies of itself. The damage that can occur due to the viruses disturbs the functionality of computers and the entire network, and steal information [6]. Ransomware is a type of malware, has the ability of preventing the legitimate user from accessing the data through encryption methods until the victim pays ransom. The attacker's first aim is financial. Generally, ransomware is being distributed via social engineering and phishing. Trojan horse is any type of software that manipulates the user by hiding harmful programs and pretending to be legitimate programs. Unfortunately, the attacker tricks the victim to open an email attachment as it comes from a legitimate sender; by the time the executable file is opened; the malware is installed. Worms is similar to viruses; it has the ability to be replicated by itself and propagated throughout a network by exploiting different flaws in the system. Spyware; this malware can monitor the user's activities on a network secretly without the knowledge of the user. Has the ability of capturing keystrokes, monitoring the screen, gathering logins details, and monitoring financial and account information [7]. Bot is a type of malware that can compromise an infected system to use the computer system's resources.

Therefore, cyber professionals are constantly enhancing the existing techniques, tools, and methods to minimize this issue or even to prevent the attackers from performing malicious activities. Malware detection techniques are divided into two parts. First, basic malware detection which has the ability to identify and restrict known malware through signature-based detection, application allowlisting and checksumming. The second part is called an advanced method which includes machine learning (ML), endpoint detection and response (EDR), and endpoint protection platform (EPP). Nowadays, cyber professionals utilize the advantages that ML provides, including detecting known malware, it has the ability of detecting malwares that have not been addressed or known previously. The process of ML is based on two steps. Firstly, extracting features like API calls, N-gram, and control flow graph from datasets that are known because they play a major role. It is not only indicating the target concept, but also speeding up the process of learning, classification, and detection. In the second process, the selection of the appropriate ML techniques is being trained such as decision tree (DT), Naïve Bayes, data mining, hidden Markov modes, and neural networks are trained for detecting and classifying the malware [8].

Malicious software is a very serious issue, and it gets complicated over the years. In fact, black hat hackers are always enhancing their skills, tools, and methods to trick and manipulate the users, and always seeking to identify flaws on the targeted machines. In case of infecting a device with malware, viruses have the ability to slow the devices and files and the entire network system can be damaged. Worms can exploit the vulnerability within the operating system (OS) including the installed programs. In addition, worms can utilize a huge amount of the system's memory resources. Attackers can encrypt credentials and sensitive data through ransomware and can deny the access of the legitimate user. Other malwares enable the attackers to steal information, credentials, and money [9].

This research is enhancing an existing technique that has the ability to anticipate malware which can impact computer devices based on Windows OS through utilizing ML algorithms. New model for malware detection in Windows OS is developed by using hybrid ML techniques; extreme gradient boosting (XGB) and random forest (RF). In addition to that, the result of the proposed model is evaluated by measuring accuracy, precision, recall, and confusion matrix [10]–[13].

2. RELATED WORK

There are different research studies on malware detection techniques. Vaidya *et al.* [14] focused on static feature extracting obtained from portable executable (PE) file and used a large number of PE. This study started with preparing the dataset, two sets of data were used which are malicious and benign executables. Malware activities can be performed such as adware, and backdoor downloaders. Then, the feature extraction from the PE which the opcode was the second phase. Opcode is an operational code that is defined as a machine language instruction that determines the operations that need to be performed. After that, the opcode was filtered to minimize features explosion, to minimize the interference of benign and malicious software and misclassification. The fourth phase was applying linear super vector machine classification and dividing the data into training and testing. The final step was monitoring the behavior using dynamic analysis to monitor the behavior of programs. The highest result achieved in this study was 95%.

Zhang *et al.* [15] used four ML models for building static malware type classifiers on PE-format files and used a recently released dataset for Windows malware detection in addition to relabeling into multi-class through VirusTotal and considered different efficient and scalable ML models. The flow work of the research was divided into three blocks: data preprocessing which is the process coming from the PE file format and VirusTotal scan reports database to feature vectors and multi-class labels. The second block is

model selection and finally, model evaluation. The researchers extracted the features of the PE file format and divided the features into a couple of categories which are file-format agnostic features and parsed PE features and each category of feature is divided into different groups. Then, data labeling was done by the researchers because the sample of the PE file format that is labeled with malicious or benign did not meet the requirement of the work and relabeling the malicious sample was necessary. The final step is the work of ML where two linear models and two ensemble DT models are used to establish the malware classifiers. The linear models that were considered known by linear support vector classifier (SVC) and logistic regression (LR), and the other two ensemble DT models are, RF and an efficient gradient boosting (GB) DT named light gradient boosting machine (LightGBM). The results of the research showed that the best model was RF because it achieved high performance with micro average F1 scored 0.96 and macro average F1 scored 0.89.

Choudhary and Sharma [16] focused on obtaining the behavioral pattern that indicates whether the software is malware which achieved through dynamic or static analysis and then afterward using ML which are super vector machine, k-nearest neighbor (KNN), Naïve Bayes, J48 DT, and multi-layer perceptron which is a deep artificial neural network. The research started with data acquisition, gathered 4,267 programs divided into 1,001 clean and 3,266 malicious programs. Then the next phase was information preprocessing and feature selection from the header of the PE file, additionally to string sequence and sequence of bytes. It is noticed that the best result was achieved by the research was 96.8%.

Wu *et al.* [17] focused on two parts, first, reinforcement learning algorithms to generate malware that can bypass detection systems with help of gym-plus. Then, based on the newly generated malwares sample, the study retrained the detection model to detect unknown threats where the accuracy results of the test of the detection of malware increased from 15.75% to 93.5%. The paper explained that in the RL model which consists of an agent and an environment. After the process of RL, the authors obtained malware samples that can evade the static PE ML malware detection model, which the aim of the work was not to attack the ML models, but to improve the ability of detection engines and anticipate variants of malicious samples and detect them in advance. The work retrained the experiment model, the research used the gym-plus LightGBM and not gradient boosting decision tree (GBDT) to enhance the percentage of detecting evaded malwares that occurred in the first part of the work; because it appeared that the percentage result of GBDT was not high as LightGBM.

The approach of Ninyesiga and Ngubiri [18] was detecting malwares through a file. The authors used 552 Windows PE with their corresponding API calls. Through Windows 7 virtual environment PE was extracted. Additionally, to the 4-gram API calls where the features were extracted using term frequency-inverse document frequency (TF-IDF). The study used benign PE files that were obtained and extracted from a freshly installed Windows OS machine. Afterwards, the classification of the study was data mining using four different classification approaches which are support vector machine (SVM), Gaussian Naïve Bayes, RF, and DT. The accuracy that detected malwares was 92% achieved by Gaussian Naïve Bayes, RF reached accuracy 95%, and both SVM and DT achieved 96.4%.

Khalid *et al.* [19] focus on fileless malware, the life cycle, and its infection chain. The study proposed fileless malware detection techniques using ML based on feature analysis. The first step of the research was to use memory forensic techniques for extracting the features representative of the fileless malware from the main memory of the system and use ML for predicting the output. The goal of using the combination of memory forensic techniques along with ML for detecting fileless malware is a promising approach because it has the ability of detecting malware that may not leave any trace on the hard disk. The authors used different datasets and MLs to accomplish the research which are VirusShare published in 2011, AnyRun published in 2016, PolySwarm published in 2018, HatchingTriage, and JoESadbox, and the ML's that are used in this research were RF, DT, SVM, LR, KNN, XGB, and GB. The first step of the research used memory forensic since it can be an effective method of detecting fileless malware which involves analyzing the contents of a computer's memory, this is also called a memory dump. The memory forensic is used to identify and extract evidence of malicious activity and after capturing the infected machine and a memory dump, then a memory forensics tool can be used like Volatility to extract the fileless malware's features, train, and test a ML model. This ML model is used to detect fileless malware on a system. In the first phase of the research that is known as acquisition of memory dump from the virtual machine, authors used VMWare Workstation 16 that are running on Windows 10 and Windows 7 to develop the research. In the next phase, which is the implementation of ML, authors used VMWare Workstation 16 by setting up a virtual machine running Windows 7. Analysis of the research was based on 45 samples and each of the samples has 33 dimensions. Dataset was divided into train and test, the research used 67% of the randomly chosen samples for training the classifiers individually and the remaining 33% samples used for testing those pre-trained classifiers. Then, two main phases occurred in the study, first feature scaling, and parameter optimization. In the feature scaling both StandardScaler and MinMaxScaler were used for scaling the features of a dataset. Considering the characteristics of the data and the specific requirements of the ML algorithm is important when choosing a method for scaling the features. Meanwhile, parameter optimization is considered

an important step in the process of building and evaluating a classifier, because it can help in enhancing the performance and generalization of the classifier. Also, it can help in reducing the computational cost of training and evaluating the classifier. Additionally, it has the ability to prevent overfitting, which might appear when a classifier too closely fits the training data and performs poorly on new, unseen data. By optimizing the parameters, researchers could achieve a balance between fitting the training data and generalizing it to new data. As a result, the research achieved by RF accuracy of 93.33%, 87.5% by SVM, and the overall accuracy of LR was 86.7%.

Based on Rezaei and Hamze [20], the aim of the study was focusing on identifying malware programs by extracting the features from header and PE files. This study used static features. The Microsoft documentation of PE file structure has several inductive features extracted from the header and the PE file's structure. The research used three different types of ML which were RF, SVM, and KNN. The authors collected datasets of 2460 PE files, including 1230 malware samples and 1230 benign samples to perform the study. The 1230 malware samples were selected randomly, and benign samples were gathered from the Program File and System32 folders and used Windows XP machine. The proposed method started with PE file format where the PE file format contains a header, and the header contains metadata about the file itself as explained in Figure 1. Then, the feature extraction where the features were extracted based on the studies of PE header and structure of the PE file. The experiments of this study were conducted in a setting that includes the following specifications: Intel(R) Core (TM) i3- 2350M CPU @2.30GHz8 with 8GB of RAM and Debian 10 as the OS. The authors derived the result from the experiment with 95.5% accuracy.

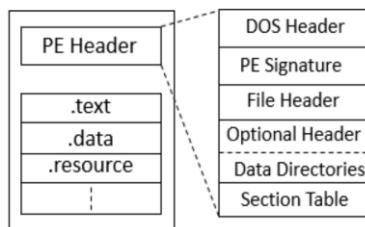


Figure 1. Features are extracted from PE file

Sharma *et al.* [21] mentioned that all the existing executable files in these days have a mutual file format known as common object file format (COFF). In addition, described that PE is a file format used by Windows for executables, object code, DLLs, and FON font files. The PE File has necessary information that helps in managing the wrapped executable code. The main two sections of PE are the header and sections. Briefly, the header holds information related to the PE file, meanwhile, the section holds the information of the executable. The goal of the study is identifying the best ML for providing the highest accuracy in detecting malware. Then, the algorithm was used to develop a web application. The method used in the research started with data preparation meaning dataset which was gathered from VirusShare which provided the latest set of malware data, Kaggle owned by Google, it is platform and is the largest community of researchers related to data science and ML, including some executable files were gathered from Windows PC. The total amount of data collection was 1000 benign files and 1000 malware files. The research divided the data, it used 70% of the data for training and 30% for testing. The next phase was featuring extraction, which was a significant step because it helped in extracting the information that was considered necessary for simplifying malware detection and classification. The researchers used Cuckoo Sandbox for feature extraction. Phase number three was, feature selection, the tree-based feature selection that uses RFs were deployed which holds several DTs. Finally, the classification that was used to develop this study was Naïve Bayes, KNN, and SVM. The accuracy of each classification, 63% were for Naïve Bayes, 91% for KNN, and 94% for SVM, this means that the height accuracy of this study was 94% by SVM.

Adamu and Awan [22] focused on a particular type of malware which ransomware. In fact, ransomware can be delivered through email attachment, drive-by download, and other vulnerability within a system. So, the methodology started with collecting dataset. Then, a feature selection technique was applied to the collected dataset which contained 30,000 attributes and used as independent variables for predicting the ransomware. However, the study found it difficult to merge all the attributes in analysis so, the authors ended with using only five attributes. The researcher utilized 942 good-ware and 582 of ransomware which belongs to 11 various ransomware family. The good-wares are represented as 0 and ransomware is represented with 1. In the research, various ML algorithms type supervised were applied known as, super

vector machine, RF, DTs, Bayesian network, artificial neural network, and LR. The training set was repeated six times during the experiment phase, and it is noticed that SVM achieved the highest accuracy result which is 88% and has lower error rate which was 0.179.

3. METHOD AND MATERIALS

In this section, model for developing an intelligent malwares detection model is proposed, which uses ML methods to determine whether there is malicious activity taking place on a device or not. The aim of the research is to investigate a combination of ML approaches to examine the possible uses of two classification models in detecting malware programs. The objective is to develop a new model that can predict if an activity is malicious or not. Determining malicious software can be considered a data mining and classification problem. Classifying the malwares can be based on behavior, characteristics such as slowing the performance of a device, a browser redirecting the user to another site that was not intended to be visited, files deleted or modified, free pop-up windows, and noticing programs starting by itself.

The proposed technique is applying multiple ML models to encourage all classifications to be able to establish a hybrid learning model. The used models are known as XGB and RF models. Therefore, this research is designing and constructing new two intelligent learning models to predict and detect malware that have the ability of destroying a device and damaging a user. Figure 2 explains the general approach of the proposed learning models for detecting malicious software.

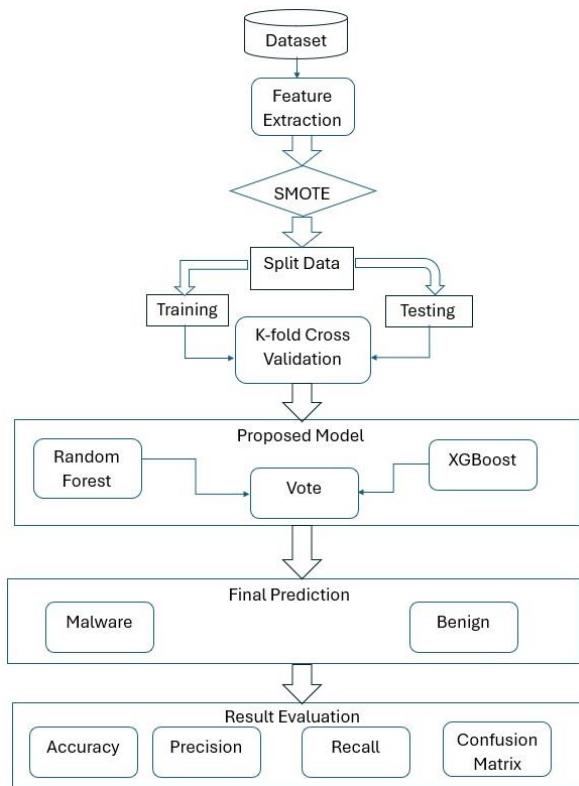


Figure 2. Overall method

The main step is encountering imbalanced data, the data could be imbalanced, so there are different techniques to deal with imbalanced data such as synthetic minority oversampling techniques (SMOTE). SMOTE is considered the common and effective method for dealing with oversampling for different domain applications. Through analyzing the data that is an existing minority class, SMOTE can be able to initiate synthetic samples. The synthetic sample has a combination of two samples from the minority class represented in linear [23].

Afterward, the most important features are selected such as exe header, checksum value, characteristics, and time data stamp that enable to differentiation between malware and benign software. The k-fold cross-validation (KCV) technique is one of the most utilized approaches by developers for model

selection and classifier error estimation. The KCV divides a dataset into k subsets, which are then used repeatedly to learn the model while the others are utilized to evaluate its performance.

3.1. Proposed model

In this section of the study, two different algorithms are being developed to deal with features that are going to be selected to distinguish between malware and benign. RF is the first selected algorithm in this research. Leo Breiman from the University of California was the first person who proposed RF. It is a collection of DTs where each tree is completely independent from another tree. Each tree is classified, and the tree "votes" for that class, in order to classify a new item based on its attributes. Based on the voting results from individual classification, the inserted test sample to the new classifier, and determining the class label of the sample can be decided. This algorithm has improved the performance of the classifier because of the random operation and can provide results in seconds because of the parallelization. It is sufficient if there is a large amount of data. Additionally, it can avoid a problem called overfitting by handling noises presented in datasets [24].

The second approach is XGB algorithm according to Zhang *et al.* [25] this is based on the GBDT and employs an ensemble learning boosting strategy for reducing the categorization of the error margin worth. The classification results of XGB are then enhanced by altering the weight of the data characteristics that were improperly categorized. In addition, the XGB approach is utilized for assessing the dependability and accuracy of algorithms for categorizing malwares. The benefit of using XGB is that it can be implemented in different applications like ranking and problem solving. Also, it is considered as a highly portable library that presently runs on OS X, Windows, and Linux platforms.

These two classifications work in parallel and individually will present the result of the prediction. Then, all of them will be voting to get the result. In the evaluation phase, which aims to evaluate the overall performance of the suggested algorithms, therefore, the study will use evaluation metrics that are most widely applied for malware detection such as classification accuracy, precision, recall, and confusion matrix.

3.2. Dataset

This part of the research describes the dataset that is going to be used in this research. Basically, the SOMLAP dataset was downloaded from Kaggle website, and it was updated at the end of 2022. This dataset applied to the PE which consists of 51,409 samples for both benign and malware files in addition to the 108 pure PE file. The sample has 19,809 malware files and 31,600 benign executable files. This paper uses the significant features that help in detecting malware based on the DOS_Header, Coff_Header or File_Header.

4. RESEARCH RESULTS AND DISCUSSION

The evaluation results of the proposed XGB algorithm and RF algorithm are discovered and explained. These results are also analyzed through using swarm optimization and ML applied to PE (SOMLAP) dataset. For assessing models' performance, a common evaluation matrix is being utilized. For classification task, this work relied on a common philosophy method evaluation known by; accuracy, precision, recall, and confusion matrix. This research used a popular website to download SOMLAP dataset which Kaggle.com which was updated at the end of 2022. This dataset applied to the PE which holds a total number of 51,409 samples for both benign and malware files in addition to the 108 pure PE file. The dataset has 19,809 malware files and 31,600 benign executable files. The first stage of the implementation phase was inspecting and data preprocessing. It was very necessary to check for missing data with the used dataset and it is being confirmed that there is not missing data within the SOMLAP dataset.

Selecting the suitable feature for models training is considered critical because selecting inappropriate features can negatively impact the performance and the final results. Therefore, applying a correlation is one of the methods that helps to determine the best features within the dataset. In addition, selecting reasonable size of features is preferable to minimize the complexity of and time consuming while focusing on the high results. Then, determine the outlier in the dataset and drop them. Once data inspection and preprocessing are completed, SMOTE is one of the methods that can solve the issue that the used dataset encounters which is imbalanced data between benign and malware samples. SMOTE helped in balancing the amount of benign and malware data. It is considered as the simplest approach to duplicate the malware sample because the dataset has 19,809 malware sample which very huge difference compared to benign samples which is 31,600 as shown in Figure 3.

Following SMOTE, KCV is utilized ML model evaluation for identifying the degree of effectiveness of the ML models are able to predict the results. This research assigned value 4 to variable K. In addition, during the implementation the transforming categorical data into numerical data must be completed because the selected features have unique values.

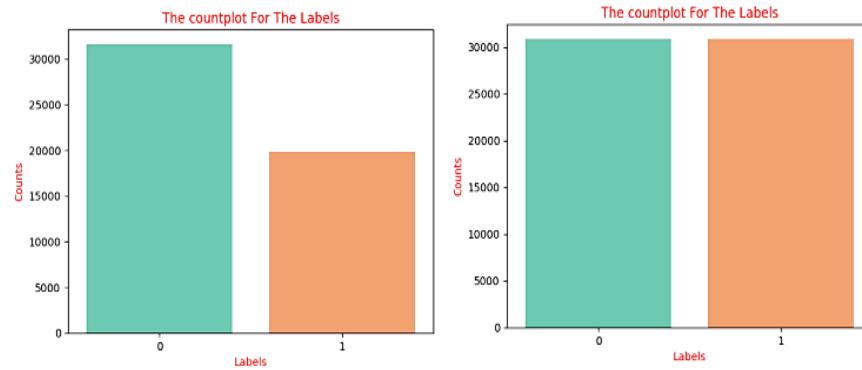


Figure 3. Before and after applying SMOTE

The next phase of this process is building models. In this case, XGB and RF are used. These models were separately run and executed at first. After that, through voting technique XGB and RF were combined for training and testing, to work hybrid and parallel as a signal once. Figures 4 to 6 are describing the results.

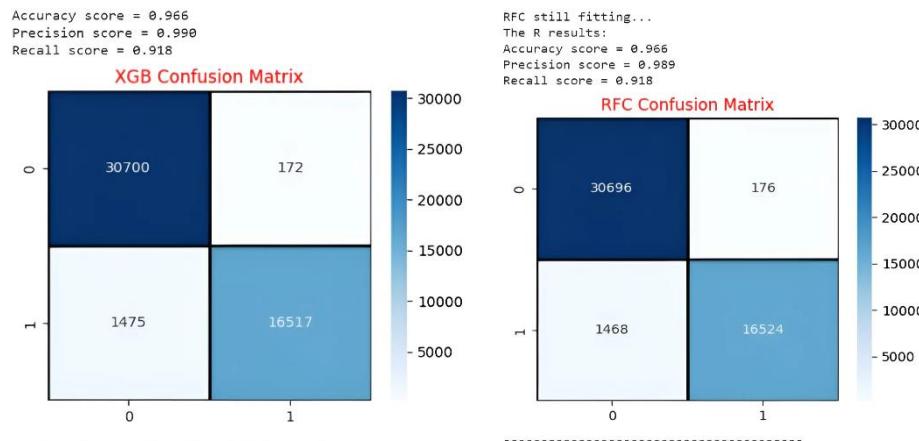


Figure 4. XGB ML results

Figure 5. RF ML results

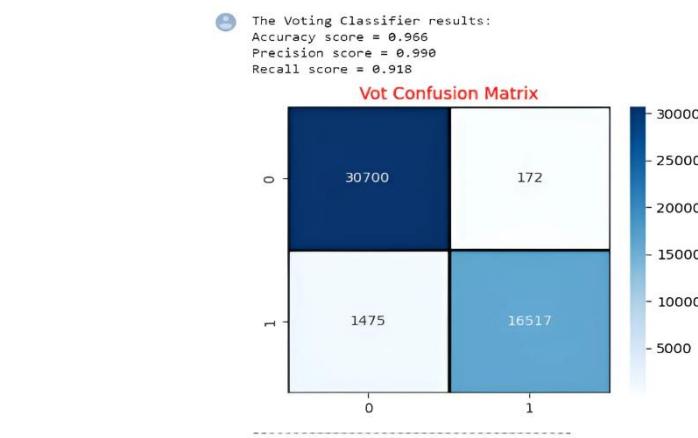


Figure 6. Voting results after combining XGB and RF

The main purpose of this research is to discover multiple ML algorithms to examine the opportunities of using two different algorithms for malware detection that can negatively impact Windows

OS. The objective of this work is to combine two types of ML algorithms to detect whether the PE file is malware or not using SOMLAP dataset. Through data mining and classification problems, malicious software can be determined and considered. Based on behavior and characteristics such as slowing the performance of a device, unintentionally redirecting a user to another website, or noticing that files are deleted or modified.

The results of the proposed model are being evaluated based on accuracy, precision, and recall. Accuracy is a popular metric used for evaluating ML algorithms. Accuracy provides a clear understanding of how often the proposed model is true. In this research the final result of the accuracy that the model predicted was 0.966. The equation of the accuracy is:

$$\text{Accuracy} = \frac{\text{Total number of correct predictions}}{\text{Total number of predictions}} \quad (1)$$

or describe as (2):

$$\text{Accuracy} = \frac{\text{True Positive} + \text{True Negative}}{\text{True Positive} + \text{True Negative} + \text{False Positive} + \text{False Negative}} \quad (2)$$

Precision is calculated by dividing the amount of real positive predictions returned by the number of correct predictions found. The equation of the precision relies on counting the amount of positive identifications that were actually true. In this work, the result that was obtained is 0.990.

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} \quad (3)$$

Measuring the recall was used to count the number of true positives that were recalled. It is helpful when the developer of the model requires us to classify events that occurred. To illustrate, if the model wants to provide detection correctly, this model must have a high recall to detect the probability. To calculate recall, the model counts actual positive predictions that were determined correctly. In this work, the result that was obtained is 0.918. Table 1 summarizes the results of accuracy, precision, and recall. Figure 7 shows the confusion matrix result.

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} \quad (4)$$

Table 1. Result discussion

Accuracy	Precision	Recall
0.966	0.990	0.918

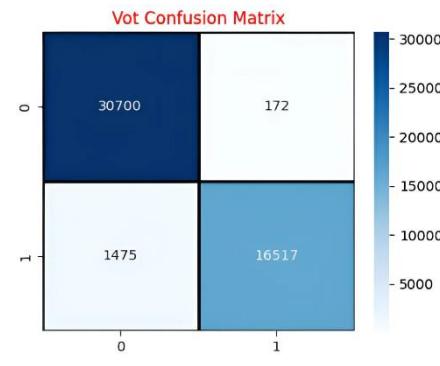


Figure 7. Confusion matrix

The confusion matrix is used to evaluate the validity of the proposed model. It provides a summary of predicted results. These results show the correct and incorrect predictions relating to the class. The confusion matrix is utilized to present significant prediction as in this work focused on accuracy, precision, recall, and confusion matrix. The benefit of the confusion matrices is providing comparisons vividly with values such as true positive, true negative, false positive, and false negative.

5. CONCLUSION

Criminals commit crimes digitally rather than committing crimes in real world, therefore, cybercriminals created malicious software to create disruptions, steal data and money and impersonate their identity and hide their geographical locations. Malwares as any type of software or program that is able to deliver and execute payload to the targeted users, systems and devices like laptop, smartphone, and servers, and these software and programs can destroy this victim. Unfortunately, there are different types of malwares that target Windows OS such as ransomware, viruses, and spyware. Additionally, hackers keep constantly making their malware complex and more disruptive. Therefore, it is very necessary to enhance the techniques and tools that can detect malware. There are different types of malware detection. Previously, the conventional methods were able to detect malware with high percentage but unfortunately, these methods are absolute and fail in detect sophisticated and complex malwares. So, nowadays with artificial intelligence and ML, it is becoming stronger and smarter in detecting malwares. ML has four different types known as supervised, unsupervised, semi-supervised, and reinforcement learning. This research aimed to detect malware that is targeting Windows OS using hybrid ML algorithms. The objective of this research was to use a comprehensive dataset and develop new models using XGB and RF. This research evaluated the results of the new model based on accuracy, precision, recall, and confusion matrix. This paper presented authors and their work who share similar aims and objectives which detecting malicious software that targets Windows OS. However, there are different accuracy results achieved by the authors. Also, this paper focused on developing XGB and RF algorithms to work as hybrid via data inspection and preprocessing at the first stage to select the optimum features and identify the correlations between the features and the class. Then splitting data is successfully completed with the help of KCV. Since the dataset was not balanced, implementing SMOTE was necessary. After that, XGB and RF algorithm is applied to be used by the voting classification to provide the final results where the final result of the accuracy achieved 0.966, precision achieved 0.990 and recall achieved 0.918. In addition, the confusion matrix was provided. The confusion matrix was used to evaluate the validity of the proposed model.

REFERENCES

- [1] S. H. Kok, A. Abdullah, N. Z. Jhanjhi, and M. Supramaniam, "Ransomware, threat and detection techniques: A review," *IJCSNS International Journal of Computer Science and Network Security*, vol. 19, no. 2, pp. 136–146, Feb. 2019.
- [2] K.-K. Kee, S. L. B. Yew, Y. S. Lim, Y. P. Ting, and R. Rashidi, "Universal cyber physical system, a prototype for predictive maintenance," *Bulletin of Electrical Engineering and Informatics*, vol. 11, no. 1, pp. 42–49, Feb. 2022, doi: 10.11591/eei.v11i1.3216.
- [3] B. Mladenov and G. Iliev, "Optimal software-defined network topology for distributed denial of service attack mitigation," *Bulletin of Electrical Engineering and Informatics*, vol. 9, no. 6, pp. 2588–2594, Dec. 2020, doi: 10.11591/eei.v9i6.2581.
- [4] A. A. Garba, M. M. Siraj, and S. H. Othman, "An assessment of cybersecurity awareness level among Northeastern University students in Nigeria," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 12, no. 1, pp. 572–584, Feb. 2022, doi: 10.11591/ijece.v12i1.pp572-584.
- [5] L. Kristiana, A. R. Darlis, and I. A. Dewi, "The feasibility of obstacle awareness forwarding scheme in a visible light communication vehicular network," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 6, pp. 6453–6460, Dec. 2020, doi: 10.11591/ijece.v10i6.pp6453-6460.
- [6] A. P. Namanya, A. Cullen, I. U. Awan, and J. P. Disso, "The world of malware: An overview," in *2018 IEEE 6th International Conference on Future Internet of Things and Cloud (FiCloud)*, Barcelona, Spain: IEEE, Aug. 2018, pp. 420–427, doi: 10.1109/FiCloud.2018.00067.
- [7] J. Singh and J. Singh, "A survey on machine learning-based malware detection in executable files," *Journal of Systems Architecture*, vol. 112, p. 101861, Jan. 2021, doi: 10.1016/j.sysarc.2020.101861.
- [8] S. K. Sahay, A. Sharma, and H. Rathore, "Evolution of malware and its detection techniques," in *Information and Communication Technology for Sustainable Development*, Singapore: Springer, 2020, pp. 139–150, doi: 10.1007/978-981-13-7166-0_14.
- [9] S. A. Roseline and S. Geetha, "A comprehensive survey of tools and techniques mitigating computer and mobile malware attacks," *Computers & Electrical Engineering*, vol. 92, p. 107143, Jun. 2021, doi: 10.1016/j.compeleceng.2021.107143.
- [10] H. Harsa *et al.*, "Machine learning and artificial intelligence models development in rainfall-induced landslide prediction," *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 12, no. 1, pp. 262–270, Mar. 2023, doi: 10.11591/ijai.v12.i1.pp262-270.
- [11] C. Sawangwong, K. Puangsawan, N. Boonnam, S. Kajornkasirat, and W. Srisang, "Classification technique for real-time emotion detection using machine learning models," *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 11, no. 4, pp. 1478–1486, Dec. 2022, doi: 10.11591/ijai.v11.i4.pp1478-1486.
- [12] A. A. Jasim, A. A. Jalal, N. M. Abdulateef, and N. A. Talib, "Effectiveness evaluation of machine learning algorithms for breast cancer prediction," *Bulletin of Electrical Engineering and Informatics*, vol. 11, no. 3, pp. 1516–1525, Jun. 2022, doi: 10.11591/eei.v11i3.3621.
- [13] T. A. Assegie, R. Subhashni, N. K. Kumar, J. P. Manivannan, P. Duraisamy, and M. F. Engidaye, "Random forest and support vector machine based hybrid liver disease detection," *Bulletin of Electrical Engineering and Informatics*, vol. 11, no. 3, pp. 1650–1656, Jun. 2022, doi: 10.11591/eei.v11i3.3787.
- [14] A. Vaidya, M. Pande, S. Shankrod, T. Dorkar, and S. Aundhakar, "Malware detection and cyber security," *International Research Journal of Modernization in Engineering Technology and Science*, vol. 4, no. 10, pp. 1584–1586, Nov. 2022, doi: 10.56726/IRJMETS31626.
- [15] S.-H. Zhang, C.-C. Kuo, and C.-S. Yang, "Static PE malware type classification using machine learning techniques," in *2019 International Conference on Intelligent Computing and its Emerging Applications (ICEA)*, Tainan, Taiwan: IEEE, Aug. 2019, pp. 81–86, doi: 10.1109/ICEA.2019.8858297.

- [16] S. Choudhary and A. Sharma, "Malware detection & classification using machine learning," in *2020 International Conference on Emerging Trends in Communication, Control and Computing (ICONC3)*, Lakshmangarh, India: IEEE, Feb. 2020, pp. 1–4, doi: 10.1109/ICONC345789.2020.9117547.
- [17] C. Wu, J. Shi, Y. Yang, and W. Li, "Enhancing machine learning based malware detection model by reinforcement learning," in *Proceedings of the 8th International Conference on Communication and Network Security*, New York, NY, USA: ACM, Nov. 2018, pp. 74–78, doi: 10.1145/3290480.3290494.
- [18] A. Ninyesiga and J. Ngubiri, "Malware classification using API system calls," *International Journal of Technology and Management*, vol. 3, no. 2, 2018.
- [19] O. Khalid *et al.*, "An insight into the machine-learning-based fileless malware detection," *Sensors*, vol. 23, no. 2, p. 612, Jan. 2023, doi: 10.3390/s23020612.
- [20] T. Rezaei and A. Hamze, "An efficient approach for malware detection using PE header specifications," in *2020 6th International Conference on Web Research (ICWR)*, Tehran, Iran: IEEE, Apr. 2020, pp. 234–239, doi: 10.1109/ICWR49608.2020.9122312.
- [21] A. Sharma, S. Mohanty, and M. R. Islam, "An experimental analysis on malware detection in executable files using machine learning," in *2021 8th International Conference on Smart Computing and Communications (ICSCC)*, Kochi, India: IEEE, Jul. 2021, pp. 178–182, doi: 10.1109/ICSCC51209.2021.9528122.
- [22] U. Adamu and I. Awan, "Ransomware prediction using supervised learning algorithms," in *2019 7th International Conference on Future Internet of Things and Cloud (FiCloud)*, Istanbul, Turkey: IEEE, Aug. 2019, pp. 57–63, doi: 10.1109/FiCloud.2019.00016.
- [23] A. J. Mohammed, "Improving classification performance for a novel imbalanced medical dataset using SMOTE method," *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 9, no. 3, pp. 3161–3172, Jun. 2020, doi: 10.30534/ijatcse/2020/104932020.
- [24] A. Parmar, R. Katariya, and V. Patel, "A review on random forest: An ensemble classifier," in *International Conference on Intelligent Data Communication Technologies and Internet of Things (ICICI) 2018*, 2019, pp. 758–763, doi: 10.1007/978-3-030-03146-6_86.
- [25] R. Zhang, B. Li, and B. Jiao, "Application of XGboost algorithm in bearing fault diagnosis," *IOP Conference Series: Materials Science and Engineering*, vol. 490, p. 072062, Apr. 2019, doi: 10.1088/1757-899X/490/7/072062.

BIOGRAPHIES OF AUTHORS



Rawabi Hilabi was born in October 1993 in Saudi Arabia. She obtained her bachelor degree in 2018 from Prince Mohammed bin Fahad University in Computer Engineering. She accomplished her Master's degree in Cybersecurity with second class honor in 2023 from Saudi Electronic University in collaboration with Colorado State University. She is Security+ certified from CompTIA. She works at Technip Energies as Instrumentation Control Systems and Automation Engineer since 2020. She can be contacted at email: Rawabi.h.hilabi@gmail.com.



Ahmed Abu-Khadrah was born in United Arab Emirates in 1981. He received Bachelor of Engineering in Computer Engineering from Alblqa Applied University in 2003. He received the master's degree in Electronic Engineering (Computer Engineering) from Universiti Teknikal Malaysia Melaka (UTeM) in 2013. He received a Ph.D. in computer engineering and communications from Universiti Teknikal Malaysia Melaka (UTeM) in 2017. He is currently Associate Professor at Faculty of Computing and Informatics at the Saudi Electronic University. His research interests in wireless network protocols, networking, communications, wireless mathematical model, and multimedia service over the networks. He can be contacted at email: abosuliman2@yahoo.com.